**Web Mining**

**Lab Assignment**

Name: **Kritika Mishra**

Registration Number: **16BCI0041**

**Question:** Use scrapy or Beautiful Soup to crawl any one of the E-commerce website of your choice and perform the same.

The following information needs to be extracted from the page:

(Choose any one product: e.g laptop, Smartphone … etc)

1. Product Name
2. Product price
3. Product discount
4. Product image

Code:

|  |  |
| --- | --- |
|  | import scrapy  from scrapy.crawler import CrawlerProcess |
|  | from scrapy.utils.log import configure\_logging |
|  | from scrapy.utils.project import get\_project\_settings |
|  | import os |
|  | import shutil |
|  | import glob |
|  |  |
|  |  |
|  |  |
|  | curfilePath = os.path.abspath(\_\_file\_\_) |
|  | curDir = os.path.abspath(os.path.join(curfilePath, os.pardir)) |
|  | tmpDir = os.path.abspath(os.path.join(curDir,'tmp/')) |
|  |  |
|  |  |
|  | # remove old crawling data |
|  | try: |
|  | shutil.rmtree(tmpDir) |
|  | except: |
|  | pass |
|  |  |
|  | # Get the search keyword from the user |
|  | print "Enter Search Keyword (product or brand name to search):", |
|  | pruduct = raw\_input() |
|  |  |
|  | # configure logging |
|  | configure\_logging({'LOG\_FORMAT': '%(levelname)s: %(message)s'}) |
|  |  |
|  | # get the project settings |
|  | s=get\_project\_settings() |
|  |  |
|  | # Change the depth limit here |
|  | # s['DEPTH\_LIMIT'] = 2 |
|  | process = CrawlerProcess(s) |
|  |  |
|  | # Add spiders to crawl |
|  | process.crawl('amazon',product=pruduct) |
|  | process.crawl('ebay',product=pruduct) |
|  | process.crawl('shopclues',product=pruduct) |
|  | process.crawl('olx',product=pruduct) |
|  |  |
|  |  |
|  | process.start() |
|  |  |
|  | # Add results to results.csv file after crawling is complete |
|  | interesting\_files = glob.glob(tmpDir+'/\*.csv') |
|  | header\_saved = False |
|  | with open('results.csv','wb') as fout: |
|  | for filename in interesting\_files: |
|  | if os.path.getsize(filename) > 0: |
|  | with open(filename) as fin: |
|  | header = next(fin) |
|  | if not header\_saved: |
|  | fout.write(header) |
|  | header\_saved = True |
|  | for line in fin: |
|  | fout.write(line) |
|  |  |
|  |  |
|  | print 'Crawling Completed' |

